

Instruction Opcode Encoding and Syntax

Revision 1.4 / 5/02/02

Instruction Class	GAS SYNTAX	Prefixes				OPCODE	NOTES
		O16	A32	OVR	REX		
CALL	CALL *.(%RIP)	--	--	--	--	FF 15 FA FF FF FF	
	CALLI *%eax	--	--	--	--	ILLEGAL	32-bit data size not allowed
	CALLq *%rax	--	--	--	--	FF D0	
	CALLI *(%ax)	--	--	--	--	ILLEGAL	32-bit data size not allowed
	CALLI *(%eax)	--	--	--	--	ILLEGAL	32-bit data size not allowed
	CALLI *(%r8)	--	--	--	--	ILLEGAL	32-bit data size not allowed
	CALLI *(%rax)	--	--	--	--	ILLEGAL	32-bit data size not allowed
	CALLq *(%ax)	--	--	--	--	ILLEGAL	32-bit data size not allowed
	CALLq *(%eax)	--	67	--	--	FF 10	A32 override: Addr64 = ZEXT(Addr32)
	CALLq *(%r8)	--	--	--	41	FF 10	REX to access upper reg.
	CALLq *(%rax)	--	--	--	--	FF 10	
	CALLw *%fs:(0)	66	--	64	--	FF 14 25 00 00 00 00	O16 Addr64 = ZEXT(Addr16) OVR FS segment override
	CALLI *%fs:(0)	--	--	64	--	64 FF 14 25 00 00 00 00	OVR FS segment override Addr64 = ZEXT(Addr32)
	CALLw *%ax	66	--	--	--	FF D0	O16 Addr64 = ZEXT(Addr16)
	CALLw *(%ax)	--	--	--	--	ILLEGAL	no 16-bit addressing
	CALLw *(%eax)	66	67	--	--	FF 10	A32 override: (Addr64) = ZEXT(Addr32) O16 Addr64 = ZEXT(Addr16)
	CALLw *(%r8)	66	--	--	41	FF 10	O16 Addr64 = ZEXT(WORD Addr16) REX to access upper reg.
	CALLw *(%rax)	66	--	--	--	FF 10	O16 Addr64 = ZEXT(Addr16)
	CALLI *%eax	--	--	--	--	FF D0	
	CALLI *(%eax)	--	67	--	--	FF 10	A32 override: (Addr64) = ZEXT(Addr32)
	CALLI *(%r8)	--	--	--	41	FF 10	REX to access upper reg.
	CALLI *(%rax)	--	--	--	--	FF 10	
	CALLq *%rax	--	--	--	--	FF D0	
	CALLq *%r8	--	--	--	41	FF D0	REX to access upper reg.
	CALLq *(%eax)	--	67	--	--	FF 10	A32 override: (Addr64) = ZEXT(Addr32)
	CALLq *(%r8)	--	--	--	41	FF 10	REX to access upper reg.
	CALLq *(%rax)	--	--	--	--	FF 10	
RET	RET	--	--	--	--	C3	
	RETw \$0x7000	--	--	--	--	C2 00 70	

RETF	--	--	--	--	CB	
RETF 0	--	--	--	--	CB	
RETN	--	--	--	--	C3	
RETN 0	--	--	--	--	C3	
RETQ	--	--	--	--	C3	
RETQ 0	--	--	--	--	C3	

IRET

IRET	--	--	--	--	CF	32-bit operand size
IRETW	66	--	--	--	CF	O16 for 16-bit operand size
IRETD	--	--	--	--	CF	32-bit operand size
IRETQ	--	--	--	48	CF	REX for 64-bit operand size

General CMP

CMPb %al,(%eax)	--	67	--	--	38 00	A32 override: Addr64 = ZEXT(Addr32)
CMPb %al,(%r8)	--	--	--	41	38 00	REX to access upper reg.
CMPb %al,(%rax)	--	--	--	--	38 00	
CMPb %al,%al	--	--	--	--	38 C0	
CMPw %ax,(%eax)	66	67	--	--	39 00	A32 override: Addr64 = ZEXT(Addr32) O16 to compare words.
CMPw %ax,(%r8)	66	--	--	41	39 00	REX to access upper reg. O16 to compare words
CMPw %ax,(%rax)	66	--	--	--	39 00	O16 to compare words.
CMPw %ax,%ax	66	--	--	--	39 C0	O16 to compare words.
CMPl %eax,(%eax)	--	67	--	--	39 00	A32 override: Addr64 = ZEXT(Addr32)
CMPl %eax,(%r8)	--	--	--	41	39 00	REX to access upper reg.
CMPl %eax,(%rax)	--	--	--	--	39 00	
CMPl %eax,%eax	--	--	--	--	3B C0	
CMPb (%eax),%al	--	67	--	--	3A 00	A32 override: Addr64 = ZEXT(Addr32)
CMPb (%r8),%al	--	--	--	41	3A 00	REX to access upper reg.
CMPb (%rax),%al	--	--	--	--	3A 00	
CMPb %al,%al	--	--	--	--	38 C0	
CMPw (%eax),%ax	66	67	--	--	3B 00	A32 override: Addr64 = ZEXT(Addr32) O16 to compare words.
CMPw (%r8),%ax	66	--	--	41	3B 00	REX to access upper reg. O16 to compare words

MOV

MOVw %cs,(%eax)	66	67	--	--	8C 08	O16 for 16-bit operand size A32 override: (Addr64) = ZEXT(Addr32)
MOVw %cs,(%r8)	66	--	--	41	8C 08	REX to access upper reg. O16 for 16-bit operand size
MOVw %cs,(%rax)	66	--	--	--	8C 08	O16 for 16-bit operand size

MOVw %ss,(%eax)	66	67	--	--	8C 10	O16 for 16-bit operand size A32 override: (Addr64) = ZEXT(Addr32)
MOVw %ss,(%r8)	66	--	--	41	8C 10	REX to access upper reg. O16 for 16-bit operand size
MOVw %ss,(%rax)	66	--	--	--	8C 10	O16 for 16-bit operand size
MOVw %fs,(%eax)	66	67	--	--	8C 20	O16 for 16-bit operand size A32 override: (Addr64) = ZEXT(Addr32)
MOVw %fs,(%r8)	66	--	--	41	8C 20	REX to access upper reg. O16 for 16-bit operand size
MOVw %fs,(%rax)	66	--	--	--	8C 20	O16 for 16-bit operand size
MOVw %cs,%ax	66	--	--	--	8C C8	O16 for 16-bit operand size
MOVw %ss,%ax	66	--	--	--	8C D0	O16 for 16-bit operand size
MOVw %fs,%ax	66	--	--	--	8C E0	O16 for 16-bit operand size
MOVI %cs,%eax	--	--	--	--	8C C8	
MOVI %cs,(%eax)	--	67	--	--	8C 08	A32 override: (Addr64) = ZEXT(Addr32)
MOVI %cs,(%r8)	--	--	--	41	8C 08	REX to access upper reg.
MOVI %cs,(%rax)	--	--	--	--	8C 08	
MOVI %ss,%eax	--	--	--	--	8C D0	
MOVI %ss,(%eax)	--	67	--	--	8C 10	A32 override: (Addr64) = ZEXT(Addr32)
MOVI %ss,(%r8)	--	--	--	41	8C 10	REX to access upper reg.
MOVI %ss,(%rax)	--	--	--	--	8C 10	
MOVI %fs,%eax	--	--	--	--	8C E0	
MOVI %fs,(%eax)	--	67	--	--	8C 20	A32 override: (Addr64) = ZEXT(Addr32)
MOVI %fs,(%r8)	--	--	--	41	8C 20	REX to access upper reg.
MOVI %fs,(%rax)	--	--	--	--	8C 20	
MOVI (%eax),%ss	66	67	--	--	8E 10	O16 for 16-bit operand size A32 override: (Addr64) = ZEXT(Addr32)
MOVI (%r8),%ss	66	--	--	41	8E 10	REX to access upper reg. O16 for 16-bit operand size
MOVI (%rax),%ss	66	--	--	--	8E 10	O16 for 16-bit operand size
MOVI (%eax),%fs	66	67	--	--	8E 20	O16 for 16-bit operand size A32 override: (Addr64) = ZEXT(Addr32)
MOVI (%r8),%fs	66	--	--	41	8E 20	REX to access upper reg. O16 for 16-bit operand size
MOVI (%rax),%fs	66	--	--	--	8E 20	O16 for 16-bit operand size
MOVI %ax,%fs	--	--	--	--	8E E0	
MOVI %ax,%ss	--	--	--	--	8E D0	
MOVI %eax,%ss	--	--	--	--	8E D0	

MOVI (%eax),%ss	--	67	--	--	8E 10	A32 override: (Addr64) = ZEXT(Addr32)
MOVI (%r8),%ss	--	--	--	41	8E 10	REX to access upper reg.
MOVI (%rax),%ss	--	--	--	--	8E 10	
MOVI %eax,%fs	--	--	--	--	8E E0	
MOVI (%eax),%fs	--	67	--	--	8E 20	A32 override: (Addr64) = ZEXT(Addr32)
MOVI (%r8),%fs	--	--	--	41	8E 20	REX to access upper reg.
MOVI (%rax),%fs	--	--	--	--	8E 20	
MOVB %fs:0,%al	--	--	64	--	8A 04 25 00 00 00 00	OVR FS segment override
MOVW %fs:0,%ax	66	--	64	--	8B 04 25 00 00 00 00	O16 for 16-bit operand size OVR FS segment override
MOVI %fs:0,%eax	--	--	64	--	8B 04 25 00 00 00 00	OVR FS segment override
MOVQ %fs:0,%rax	--	--	64	48	8B 04 25 00 00 00 00	REX for 64-bit operand size OVR FS segment override
MOVB %al,%fs:0	--	--	64	--	88 04 25 00 00 00 00	OVR FS segment override
MOVW %ax,%fs:0	66	--	64	--	89 04 25 00 00 00 00	O16 for 16-bit operand size OVR FS segment override
MOVI %eax,%fs:0	--	--	64	--	89 04 25 00 00 00 00	OVR FS segment override
MOVQ %rax,%fs:0	--	--	64	48	89 04 25 00 00 00 00	REX for 64-bit operand size OVR FS segment override
MOVQ %cr4,%rax	--	--	--	--	0F 20 E0	
MOVQ %cr4,%r8	--	--	--	41	0F 20 E0	REX for 64-bit operand size REX to access upper reg.
MOVQ %cr0,%rax	--	--	--	--	0F 20 C0	
MOVQ %cr15,%r8	--	--	--	45	0F 20 F8	REX for 64-bit operand size REX to access upper reg.
MOVQ %cr15,%rax	--	--	--	44	0F 20 F8	REX for 64-bit operand size REX to access upper reg.
MOVQ %cr0,%r8	--	--	--	41	0F 20 C0	REX for 64-bit operand size REX to access upper reg.
MOVQ %dr0,%rax	--	--	--	--	0F 21 C0	
MOVQ %dr15,%r8	--	--	--	45	0F 21 F8	REX for 64-bit operand size REX to access upper reg.
MOVQ %dr15,%rax	--	--	--	44	0F 21 F8	REX for 64-bit operand size REX to access upper reg.
MOVQ %dr0,%r8	--	--	--	41	0F 21 C0	REX for 64-bit operand size REX to access upper reg.
MOVI %tr3,%eax	--	--	--	--	0F 24 D8	
MOVQ %rax,%cr4	--	--	--	--	0F 22 E0	
MOVQ %r8,%cr4	--	--	--	41	0F 22 E0	REX for 64-bit operand size REX to access upper reg.

MOVq %rax,%cr0	--	--	--	--	0F 22 C0	
MOVq %r8,%cr15	--	--	--	45	0F 22 F8	REX for 64-bit operand size REX to access upper reg.
MOVq %r8,%cr0	--	--	--	41	0F 22 C0	REX for 64-bit operand size REX to access upper reg.
MOVq %rax,%cr15	--	--	--	44	0F 22 F8	REX for 64-bit operand size REX to access upper reg.
MOVq %rax,%dr0	--	--	--	--	0F 23 C0	
MOVq %r8,%dr15	--	--	--	45	0F 23 F8	REX for 64-bit operand size REX to access upper reg.
MOVq %r8,%dr0	--	--	--	41	0F 23 C0	REX for 64-bit operand size REX to access upper reg.
MOVq %rax,%dr15	--	--	--	44	0F 23 F8	REX for 64-bit operand size REX to access upper reg.
MOVI %eax,%tr3	--	--	--	--	0F 26 D8	
MOVb %al,(%eax)	--	67	--	--	88 00	A32 override: (Addr64) = ZEXT(Addr32)
MOVb %al,(%r8)	--	--	--	41	88 00	REX to access upper reg.
MOVb %al,(%rax)	--	--	--	--	88 00	
MOVb %al,%al	--	--	--	--	8A C0	
MOVw %ax,(%eax)	66	67	--	--	89 00	O16 for 16-bit operand size A32 override: (Addr64) = ZEXT(Addr32)
MOVw %ax,(%r8)	66	--	--	41	89 00	REX to access upper reg. O16 for 16-bit operand size
MOVw %ax,(%rax)	66	--	--	--	89 00	O16 for 16-bit operand size
MOVw %ax,%ax	66	--	--	--	8B C0	O16 for 16-bit operand size
MOVI %eax,(%eax)	--	67	--	--	89 00	A32 override: (Addr64) = ZEXT(Addr32)
MOVI %eax,(%r8)	--	--	--	41	89 00	REX to access upper reg.
MOVI %eax,(%rax)	--	--	--	--	89 00	
MOVI %eax,%eax	--	--	--	--	8B C0	
MOVb (%eax),%al	--	67	--	--	8A 00	A32 override: (Addr64) = ZEXT(Addr32)
MOVb (%r8),%al	--	--	--	41	8A 00	REX to access upper reg.
MOVb (%rax),%al	--	--	--	--	8A 00	
MOVb %al,%al	--	--	--	--	8A C0	
MOVw (%eax),%ax	66	67	--	--	8B 00	O16 for 16-bit operand size A32 override: (Addr64) = ZEXT(Addr32)
MOVw (%r8),%ax	66	--	--	41	8B 00	REX to access upper reg. O16 for 16-bit operand size
MOVw (%rax),%ax	66	--	--	--	8B 00	O16 for 16-bit operand size
MOVw %ax,%ax	66	--	--	--	89 C0	O16 for 16-bit operand size
MOVI (%eax),%eax	--	67	--	--	8B 00	A32 override: (Addr64) = ZEXT(Addr32)

MOVI (%r8),%eax	--	--	--	41	8B 00	REX to access upper reg.
MOVI (%rax),%eax	--	--	--	--	8B 00	
MOVI %eax,%eax	--	--	--	--	8B C0	
MOVb \$0,%al	--	--	--	--	B0 00	
MOVw \$0x7000,%ax	66	--	--	--	B8 00 70	O16 for 16-bit operand size
MOVI \$0x70000000,%eax	--	--	--	--	B8 00 00 00 70	
MOVq 0,%rax	--	--	--	48	B8 00 00 00 00 00 00 00 00	REX for 64-bit operand size
MOVq 0,%r8	--	--	--	49	B8 00 00 00 00 00 00 00 00	REX for 64-bit operand size REX to access upper reg.
MOVb \$0,%al	--	--	--	--	B0 00	
MOVb \$0,(%eax)	--	67	--	--	C6 00 00	A32 override: (Addr64) = ZEXT(Addr32)
MOVb \$0,(%r8)	--	--	--	41	C6 00 00	REX to access upper reg.
MOVb \$0,(%rax)	--	--	--	--	C6 00 00	
MOVw \$0x7000,%ax	66	--	--	--	B8 00 70	O16 for 16-bit operand size
MOVw \$0x7000,(%eax)	66	67	--	--	C7 00 00 70	O16 for 16-bit operand size A32 override: (Addr64) = ZEXT(Addr32)
MOVw \$0x7000,(%r8)	66	--	--	41	C7 00 00 70	REX to access upper reg. O16 for 16-bit operand size
MOVw \$0x7000,(%rax)	66	--	--	--	C7 00 00 70	O16 for 16-bit operand size
MOVI \$0x70000000,%eax	--	--	--	--	B8 00 00 00 70	
MOVI \$0x70000000,(%eax)	--	67	--	--	C7 00 00 00 00 70	A32 override: (Addr64) = ZEXT(Addr32)
MOVI \$0x70000000,(%r8)	--	--	--	41	C7 00 00 00 00 70	REX to access upper reg.
MOVI \$0x70000000,(%rax)	--	--	--	--	C7 00 00 00 00 70	
MOVq \$0x7000000000000000,%rax	--	--	--	48	B8 00 00 00 00 00 00 00 70	REX for 64-bit operand size
MOVq \$0x7000000000000000,%r8	--	--	--	49	B8 00 00 00 00 00 00 00 70	REX for 64-bit operand size REX to access upper reg.
MOVb \$0,(%eax)	--	67	--	--	C6 00 00	
MOVb \$0,(%r8)	--	--	--	41	C6 00 00	REX to access upper reg.
MOVb \$0,(%rax)	--	--	--	--	C6 00 00	
MOVw \$0x7000,(%eax)	66	67	--	--	C7 00 00 70	O16 for 16-bit operand size A32 override: (Addr64) = ZEXT(Addr32)

MOVw \$0x7000,(%r8)	66	--	--	--	41 C7 00 00 70	O16 for 16-bit operand size
MOVw \$0x7000,(%rax)	66	--	--	--	C7 00 00 70	O16 for 16-bit operand size
MOVI \$0x70000000,(%eax)	--	67	--	--	C7 00 00 00 00 70	A32 override: (Addr64) = ZEXT(Addr32)
MOVI \$0x70000000,(%r8)	--	67	--	--	C7 00 00 00 00 70	A32 override: (Addr64) = ZEXT(Addr32)
MOVI \$0x70000000,(%rax)	--	--	--	--	C7 00 00 00 00 70	
MOVB \$0,(%eax)	--	67	--	--	C6 00 00	A32 override: (Addr64) = ZEXT(Addr32)
MOVB \$0,(%r8)	--	--	--	41	C6 00 00	REX to access upper reg.
MOVB \$0,(%rax)	--	--	--	--	C6 00 00	
MOVw \$0x7000,(%eax)	66	67	--	--	C7 00 00 70	O16 for 16-bit operand size A32 override: (Addr64) = ZEXT(Addr32)
MOVw \$0x7000,(%r8)	66	--	--	41	C7 00 00 70	REX to access upper reg. O16 for 16-bit operand size
MOVw \$0x7000,(%rax)	66	--	--	--	C7 00 00 70	O16 for 16-bit operand size
MOVI \$0x70000000,(%eax)	--	67	--	--	C7 00 00 00 00 70	A32 override: (Addr64) = ZEXT(Addr32)
MOVI \$0x70000000,(%r8)	--	--	--	41	C7 00 00 00 00 70	REX to access upper reg.
MOVI \$0x70000000,(%rax)	--	--	--	--	C7 00 00 00 00 70	
MOVq %rax,%rax	--	--	--	48	8B C0	REX for 64-bit operand size
MOVq %r8,%r8	--	--	--	4D	89 C0	REX for 64-bit operand size REX to access upper reg.
MOVq %r8,%rax	--	--	--	4C	89 C0	REX for 64-bit operand size REX to access upper reg.
MOVq %rax,%r8	--	--	--	49	89 C0	REX for 64-bit operand size REX to access upper reg.
MOVq %rax,(%eax)	--	67	--	48	89 00	REX for 64-bit operand size A32 override: (Addr64) = ZEXT(Addr32)
MOVq %rax,(%r8)	--	--	--	49	89 00	REX for 64-bit operand size REX to access upper reg.
MOVq %rax,(%rax)	--	--	--	48	89 00	REX for 64-bit operand size
MOVq %r8,(%eax)	--	67	--	4C	89 00	REX for 64-bit operand size REX to access upper reg. A32 override: (Addr64) = ZEXT(Addr32)
MOVq %r8,(%r8)	--	--	--	4D	89 00	REX for 64-bit operand size REX to access upper reg.
MOVq %r8,(%rax)	--	--	--	4C	89 00	REX for 64-bit operand size REX to access upper reg.

MOVq (%eax),%rax	--	67	--	48	8B 00	REX for 64-bit operand size A32 override: (Addr64) = ZEXT(Addr32)
MOVq (%r8),%rax	--	--	--	49	8B 00	REX for 64-bit operand size REX to access upper reg.
MOVq (%rax),%rax	--	--	--	48	8B 00	REX for 64-bit operand size
MOVq (%eax),%r8	--	67	--	4C	8B 00	REX for 64-bit operand size REX to access upper reg. A32 override: (Addr64) = ZEXT(Addr32)
MOVq (%r8),%r8	--	--	--	4D	8B 00	REX for 64-bit operand size REX to access upper reg.
MOVq (%rax),%r8	--	--	--	4C	8B 00	REX for 64-bit operand size REX to access upper reg.
MOVq \$0x70000000,%rax	--	--	--	48	B8 00 00 00 70 00 00 00 00	REX for 64-bit operand size
MOVq \$0x70000000,%r8	--	--	--	49	B8 00 00 00 70 00 00 00 00	REX for 64-bit operand size REX to access upper reg.
MOVq \$0x70000000,(%eax)	--	67	--	48	C7 00 00 00 00 70	REX for 64-bit operand size A32 override: (Addr64) = ZEXT(Addr32)
MOVq \$0x70000000,(%r8)	--	--	--	49	C7 00 00 00 00 70	REX for 64-bit operand size REX to access upper reg.
MOVq \$0x70000000,(%rax)	--	--	--	48	C7 00 00 00 00 70	REX for 64-bit operand size

MOVNTI

MOVNTI %eax(%eax)	--	67	--	--	0f c3 00	A32 override: (Addr64) = ZEXT(Addr32)
MOVNTI %eax,(%r8)	--	--	--	41	0f c3 00	REX to access upper reg.
MOVNTI %eax,(%rax)	--	--	--	--	0f c3 00	
MOVNTI %rax,(%eax)	--	67	--	48	0f c3 00	A32 override: (Addr64) = ZEXT(Addr32) REX for 64-bit operand size
MOVNTI %rax,(%r8)	--	--	--	49	0F C3 00	REX to access upper reg. REX for 64-bit operand size
MOVNTI %rax,(%rax)	--	--	--	48	0F C3 00	REX for 64-bit operand size REX to access upper reg.
MOVNTI %r8,(%eax)	--	67	--	4C	0F C3 00	A32 override: (Addr64) = ZEXT(Addr32) REX for 64-bit operand size
MOVNTI %r8,(%r8)	--	--	--	4D	0F C3 00	REX to access upper reg. REX for 64-bit operand size
MOVNTI %r8,(%rax)	--	--	--	4C	0F C3 00	REX to access upper reg. REX for 64-bit operand size

Conditional

LOOP

LOOPq .	--	--	--	--	E2 FE	RCX used as loop counter
LOOPI .	--	67	--	--	E2 FD	ECX override by 0x67
LOOPq ?? .	66	--	--	--	E2 FE	A16 override: (Addr64) = ZEXT(Addr16) RCX used as loop counter
LOOPI ?? .	66	67	--	--	E2 FD	A16 override: (Addr64) = ZEXT(Addr16) ECX override by 0x67

Jcc

Ja .	--	--	--	--	77 FE	
Ja .+0xFFFFFFFF	--	--	--	--	0F 87 FA FF FF FF	
	66	--	--	--	77 FD	A16 override: (Addr64) = ZEXT(Addr16)
	66	--	--	--	0F 87 F9 FF FF FF	A16 override: (Addr64) = ZEXT(Addr16)

J*CXZ

JCXZ .	--	--	--	--	ILLEGAL	No prefix exists to select CX as a counter
JECXZ .	--	67	--	--	E3 FD	ECX used as counter
JRCXZ .	--	--	--	--	E3 FE	RCX used as counter
	66	67	--	--	E3 FC	ECX used as counter A16 override: (Addr64) = ZEXT(Addr16)
	66	--	--	--	E3 FD	A16 override: (Addr64) = ZEXT(Addr16)

Integer Math

IDIV

IDIVb %al	--	--	--	--	F6 F8	Sign extended result
IDIVb (%eax)	--	67	--	--	F6 38	Sign extended result A32 override: (Addr64) = ZEXT(Addr32)
IDIVb (%r8)	--	--	--	41	F6 38	Sign extended result REX to access upper reg.
IDIVb (%rax)	--	--	--	--	F6 38	Sign extended result
IDIVw %ax	66	--	--	--	F7 F8	Sign extended result O16 for 16-bit operand size
IDIVw (%eax)	66	67	--	--	F7 38	Sign extended result O16 for 16-bit operand size A32 override: (Addr64) = ZEXT(Addr32)

IDIVw (%r8)	66	--	--	41	F7 38	Sign extended result REX to access upper reg. O16 for 16-bit operand size
IDIVw (%rax)	66	--	--	--	F7 38	Sign extended result O16 for 16-bit operand size
IDIV %eax	--	--	--	--	F7 F8	Sign extended result
IDIV (%eax)	--	67	--	--	F7 38	Sign extended result A32 override: (Addr64) = ZEXT(Addr32)
IDIV (%r8)	--	--	--	41	F7 38	Sign extended result REX to access upper reg
IDIV (%rax)	--	--	--	--	F7 38	Sign extended result
IDIVq %rax	--	--	--	48	F7 F8	Sign extended result REX for 64-bit operand size
IDIVq %r8	--	--	--	49	F7 F8	Sign extended result REX for 64-bit operand size REX to access upper reg
IDIVq (%eax)	--	67	--	48	F7 38	Sign extended result REX for 64-bit operand size A32 override: (Addr64) = ZEXT(Addr32)
IDIVq (%r8)	--	--	--	49	F7 38	Sign extended result REX for 64-bit operand size REX to access upper reg
IDIVq (%rax)	--	--	--	48	F7 38	Sign extended result REX for 64-bit operand size

IMULb %al	--	--	--	--	F6 E8	Sign extended result
IMULb (%eax)	--	67	--	--	F6 28	Sign extended result A32 override: (Addr64) = ZEXT(Addr32)
IMULb (%r8)	--	--	--	41	F6 28	Sign extended result REX to access upper reg
IMULb (%rax)	--	--	--	--	F6 28	Sign extended result
IMULw %ax	66	--	--	--	F7 E8	Sign extended result O16 for 16-bit operand size
IMULw (%eax)	66	67	--	--	F7 28	Sign extended result O16 for 16-bit operand size A32 override: (Addr64) = ZEXT(Addr32)
IMULw (%r8)	66	--	--	41	F7 28	Sign extended result O16 for 16-bit operand size REX to access upper reg
IMULw (%rax)	66	--	--	--	F7 28	Sign extended result O16 for 16-bit operand size
IMUL %eax	--	--	--	--	F7 E8	Sign extended result

IMULI (%eax)	--	67	--	--	F7 28	Sign extended result A32 override: (Addr64) = ZEXT(Addr32)
IMULI (%r8)	--	--	--	41	F7 28	Sign extended result REX to access upper reg
IMULI (%rax)	--	--	--	--	F7 28	Sign extended result
IMULw (%eax),%ax	66	67	--	--	0F AF 00	Sign extended result O16 for 16-bit operand size A32 override: (Addr64) = ZEXT(Addr32)
IMULw (%r8),%ax	66	--	--	41	0F AF 00	Sign extended result REX to access upper reg. O16 for 16-bit operand size
IMULw (%rax),%ax	66	--	--	--	0F AF 00	Sign extended result O16 for 16-bit operand size
IMULw %ax,%ax	66	--	--	--	0F AF C0	Sign extended result O16 for 16-bit operand size
IMULI (%eax),%eax	--	67	--	--	0F AF 00	Sign extended result A32 override: (Addr64) = ZEXT(Addr32)
IMULI (%r8),%eax	--	--	--	41	0F AF 00	Sign extended result REX to access upper reg
IMULI (%rax),%eax	--	--	--	--	0F AF 00	Sign extended result
IMULI %eax,%eax	--	--	--	--	0F AF C0	Sign extended result
IMULq (%eax)	--	67	--	48	F7 28	Sign extended result REX for 64-bit operand size A32 override: (Addr64) = ZEXT(Addr32)
IMULq (%r8)	--	--	--	49	F7 28	Sign extended result REX for 64-bit operand size REX to access upper reg
IMULq (%rax)	--	--	--	48	F7 28	Sign extended result REX for 64-bit operand size
IMULq %rax	--	--	--	48	F7 E8	Sign extended result REX for 64-bit operand size
IMULq %r8	--	--	--	49	F7 E8	Sign extended result REX for 64-bit operand size REX to access upper reg
IMULq %rax,%rax	--	--	--	48	0F AF C0	Sign extended result REX for 64-bit operand size
IMULq %r8,%r8	--	--	--	4D	0F AF C0	Sign extended result REX for 64-bit operand size REX to access upper reg
IMULq %r8,%rax	--	--	--	49	0F AF C0	Sign extended result REX for 64-bit operand size REX to access upper reg
IMULq %rax,%r8	--	--	--	4C	0F AF C0	Sign extended result REX for 64-bit operand size REX to access upper reg

IMULq (%eax),%rax	--	67	--	48	0F AF 00	Sign extended result REX for 64-bit operand size A32 override: (Addr64) = ZEXT(Addr32)
IMULq (%r8),%rax	--	--	--	49	0F AF 00	Sign extended result REX for 64-bit operand size REX to access upper reg
IMULq (%rax),%rax	--	--	--	48	0F AF 00	Sign extended result REX for 64-bit operand size
IMULq (%eax),%r8	--	67	--	4C	0F AF 00	Sign extended result REX for 64-bit operand size REX to access upper reg. A32 override: (Addr64) = ZEXT(Addr32)
IMULq (%r8),%r8	--	--	--	4D	0F AF 00	Sign extended result REX for 64-bit operand size REX to access upper reg
IMULq (%rax),%r8	--	--	--	4C	0F AF 00	Sign extended result REX for 64-bit operand size REX to access upper reg
IMULq \$0,%rax	--	--	--	48	6B C0 00	Sign extended result REX for 64-bit operand size
IMULq \$0,%r8	--	--	--	4D	6B C0 00	Sign extended result REX for 64-bit operand size REX to access upper reg
IMULq \$0x70000000,%rax	--	--	--	48	69 C0 00 00 00 70	Sign extended result REX for 64-bit operand size
IMULq \$0x70000000,%r8	--	--	--	4D	69 C0 00 00 00 70	Sign extended result REX for 64-bit operand size REX to access upper reg
IMULq \$0,(%eax),%rax	--	67	--	48	6B 00 00	Sign extended result REX for 64-bit operand size A32 override: (Addr64) = ZEXT(Addr32)
IMULq \$0,(%r8),%rax	--	--	--	49	6B 00 00	Sign extended result REX for 64-bit operand size REX to access upper reg
IMULq \$0,(%rax),%rax	--	--	--	48	6B 00 00	Sign extended result REX for 64-bit operand size
IMULq \$0,(%eax),%r8	--	67	--	4C	6B 00 00	Sign extended result REX for 64-bit operand size REX to access upper reg. A32 override: (Addr64) = ZEXT(Addr32)
IMULq \$0,(%r8),%r8	--	--	--	4D	6B 00 00	Sign extended result REX for 64-bit operand size REX to access upper reg

IMULq \$0,(%rax),%r8	--	--	--	4C	6B 00 00	Sign extended result REX for 64-bit operand size REX to access upper reg
IMULq \$0,%rax,%rax	--	--	--	48	6B C0 00	Sign extended result REX for 64-bit operand size
IMULq \$0,%r8,%r8	--	--	--	4D	6B C0 00	Sign extended result REX for 64-bit operand size REX to access upper reg
IMULq \$0,%r8,%rax	--	--	--	49	6B C0 00	Sign extended result REX for 64-bit operand size REX to access upper reg
IMULq \$0,%rax,%r8	--	--	--	4C	6B C0 00	Sign extended result REX for 64-bit operand size REX to access upper reg
IMULq \$0x70000000,%rax,%rax	--	--	--	48	69 C0 00 00 00 70	Sign extended result REX for 64-bit operand size
IMULq \$0x70000000,%r8,%r8	--	--	--	4D	69 C0 00 00 00 70	Sign extended result REX for 64-bit operand size REX to access upper reg
IMULq \$0x70000000,%r8,%rax	--	--	--	49	69 C0 00 00 00 70	Sign extended result REX for 64-bit operand size REX to access upper reg
IMULq \$0x70000000,%rax,%r8	--	--	--	4C	69 C0 00 00 00 70	Sign extended result REX for 64-bit operand size REX to access upper reg
IMULw \$0,%ax,%ax	66	--	--	--	6B C0 00	Sign extended result O16 for 16-bit operand size
IMULw \$0,(%eax),%ax	66	67	--	--	6B 00 00	Sign extended result O16 for 16-bit operand size A32 override: (Addr64) = ZEXT(Addr32)
IMULw \$0,(%r8),%ax	66	--	--	41	6B 00 00	Sign extended result REX to access upper reg. O16 for 16-bit operand size
IMULw \$0,(%rax),%ax	66	--	--	--	6B 00 00	Sign extended result O16 for 16-bit operand size
IMULw \$0x7000,%ax,%ax	66	--	--	--	69 C0 00 70	Sign extended result O16 for 16-bit operand size
IMULw \$0x7000,(%eax),%ax	66	67	--	--	69 00 00 70	Sign extended result O16 for 16-bit operand size A32 override: (Addr64) = ZEXT(Addr32)
IMULw \$0x7000,(%r8),%ax	66	--	--	41	69 00 00 70	Sign extended result REX to access upper reg. O16 for 16-bit operand size
IMULw \$0x7000,(%rax),%ax	66	--	--	--	69 00 00 70	Sign extended result O16 for 16-bit operand size

IMULI \$0,(%eax),%eax	--	67	--	--	6B 00 00	Sign extended result A32 override: (Addr64) = ZEXT(Addr32)
IMULI \$0,(%r8),%eax	--	--	--	41	6B 00 00	Sign extended result REX to access upper reg
IMULI \$0,(%rax),%eax	--	--	--	--	6B 00 00	Sign extended result
IMULI \$0,%eax,%eax	--	--	--	--	6B C0 00	Sign extended result
IMULI \$0,(%eax),%eax	--	67	--	--	6B 00 00	Sign extended result A32 override: (Addr64) = ZEXT(Addr32)
IMULI \$0,(%r8),%eax	--	--	--	41	6B 00 00	Sign extended result REX to access upper reg
IMULI \$0,(%rax),%eax	--	--	--	--	6B 00 00	Sign extended result
IMULI \$0x70000000,%eax,%eax	--	--	--	--	69 C0 00 00 00 70	Sign extended result
IMULw \$0,%ax	66	--	--	--	6B C0 00	Sign extended result O16 for 16-bit operand size
IMULw \$0x7000,%ax	66	--	--	--	69 C0 00 70	Sign extended result O16 for 16-bit operand size
IMULI \$0,%eax	--	--	--	--	6B C0 00	Sign extended result
IMULI \$0x70000000,%eax	--	--	--	--	69 C0 00 00 00 70	Sign extended result

SIMD/SSE

ADDPD

ADDPD (%eax),%xmm0	--	67	66	--	0F 58 00	A32 override: (Addr64) = ZEXT(Addr32) OVR 128-bit media instruction override
ADDPD (%r8),%xmm0	--	--	66	41	0F 58 00	REX to access upper reg. OVR 128-bit media instruction override
ADDPD (%rax),%xmm0	--	--	66	--	0F 58 00	OVR 128-bit media instruction override
ADDPD (%eax),%xmm15	--	67	66	44	0F 58 38	REX to access upper XMM reg A32 override: (Addr64) = ZEXT(Addr32) OVR 128-bit media instruction override
ADDPD (%r8),%xmm15	--	--	66	45	0F 58 38	REX to access upper XMM reg REX to access upper reg. OVR 128-bit media instruction override
ADDPD (%rax),%xmm15	--	--	66	44	0F 58 38	REX to access upper XMM reg OVR 128-bit media instruction override
ADDPD (%eax),%xmm8	--	67	66	44	0F 58 00	REX to access upper XMM reg A32 override: (Addr64) = ZEXT(Addr32) OVR 128-bit media instruction override
ADDPD (%r8),%xmm8	--	--	66	45	0F 58 00	REX to access upper XMM reg REX to access upper reg. OVR 128-bit media instruction override
ADDPD (%rax),%xmm8	--	--	66	44	0F 58 00	REX to access upper XMM reg OVR 128-bit media instruction override

ADDPD (%eax),%xmm7	--	67	66	--	0F 58 38	A32 override: (Addr64) = ZEXT(Addr32) OVR 128-bit media instruction override
ADDPD (%r8),%xmm7	--	--	66	41	0F 58 38	REX to access upper reg. OVR 128-bit media instruction override
ADDPD (%rax),%xmm7	--	--	66	--	0F 58 38	OVR 128-bit media instruction override
ADDPD %xmm0,%xmm0	--	--	66	--	0F 58 C0	OVR 128-bit media instruction override
ADDPD %xmm15,%xmm15	--	--	66	45	0F 58 FF	REX to access upper XMM reg OVR 128-bit media instruction override
ADDPD %xmm15,%xmm8	--	--	66	45	0F 58 C7	REX to access upper XMM reg OVR 128-bit media instruction override

CMPPD

CMPPD \$0,(%eax),%xmm0	--	67	66	--	0F C2 00 00	A32 override: (Addr64) = ZEXT(Addr32) OVR 128-bit media instruction override
CMPPD \$0,(%r8),%xmm0	--	--	66	41	0F C2 00 00	REX to access upper reg. OVR 128-bit media instruction override
CMPPD \$0,(%rax),%xmm0	--	--	66	--	0F C2 00 00	OVR 128-bit media instruction override
CMPPD \$0,(%eax),%xmm15	--	67	66	44	0F C2 38 00	A32 override: (Addr64) = ZEXT(Addr32) OVR 128-bit media instruction override REX to access upper XMM reg
CMPPD \$0,(%r8),%xmm15	--	--	66	45	0F C2 38 00	REX to access upper XMM reg REX to access upper reg. OVR 128-bit media instruction override
CMPPD \$0,(%rax),%xmm15	--	--	66	44	0F C2 38 00	REX to access upper XMM reg OVR 128-bit media instruction override
CMPPD \$0,(%eax),%xmm8	--	67	66	44	0F C2 00 00	REX to access upper XMM reg A32 override: (Addr64) = ZEXT(Addr32) OVR 128-bit media instruction override
CMPPD \$0,(%r8),%xmm8	--	--	66	45	0F C2 00 00	REX to access upper XMM reg REX to access upper reg. OVR 128-bit media instruction override
CMPPD \$0,(%rax),%xmm8	--	--	66	44	0F C2 00 00	REX to access upper XMM reg OVR 128-bit media instruction override
CMPPD \$0,(%eax),%xmm7	--	67	66	--	0F C2 38 00	A32 override: (Addr64) = ZEXT(Addr32) OVR 128-bit media instruction override
CMPPD \$0,(%r8),%xmm7	--	--	66	41	0F C2 38 00	REX to access upper reg. OVR 128-bit media instruction override
CMPPD \$0,(%rax),%xmm7	--	--	66	--	0F C2 38 00	OVR 128-bit media instruction override
CMPPD \$0,%xmm0,%xmm0	--	--	66	--	0F C2 C0 00	OVR 128-bit media instruction override
CMPPD \$0,%xmm15,%xmm15	--	--	66	45	0F C2 FF 00	REX to access upper XMM reg OVR 128-bit media instruction override
CMPPD \$0,%xmm15,%xmm8	--	--	66	45	0F C2 C7 00	REX to access upper XMM reg OVR 128-bit media instruction override

CMPPD \$0,%xmm15,%xmm7	--	--	66	41	0F C2 FF 00	REX to access upper XMM reg OVR 128-bit media instruction override
CMPPD \$0,%xmm15,%xmm0	--	--	66	41	0F C2 C7 00	REX to access upper XMM reg OVR 128-bit media instruction override
CMPPD \$0,%xmm8,%xmm15	--	--	66	45	0F C2 F8 00	REX to access upper XMM reg OVR 128-bit media instruction override
CMPPD \$0,%xmm8,%xmm8	--	--	66	45	0F C2 C0 00	REX to access upper XMM reg OVR 128-bit media instruction override
CMPPD \$0,%xmm8,%xmm7	--	--	66	41	0F C2 F8 00	REX to access upper XMM reg OVR 128-bit media instruction override
CMPPD \$0,%xmm8,%xmm0	--	--	66	41	0F C2 C0 00	REX to access upper XMM reg OVR 128-bit media instruction override
CMPPD \$0,%xmm7,%xmm15	--	--	66	44	0F C2 FF 00	REX to access upper XMM reg OVR 128-bit media instruction override
CMPPD \$0,%xmm7,%xmm8	--	--	66	44	0F C2 C7 00	REX to access upper XMM reg OVR 128-bit media instruction override
CMPPD \$0,%xmm7,%xmm7	--	--	66	--	0F C2 FF 00	OVR 128-bit media instruction override
CMPPD \$0,%xmm7,%xmm0	--	--	66	--	0F C2 C7 00	OVR 128-bit media instruction override
CMPPD \$0,%xmm0,%xmm15	--	--	66	44	0F C2 F8 00	REX to access upper XMM reg OVR 128-bit media instruction override
CMPPD \$0,%xmm0,%xmm8	--	--	66	44	0F C2 C0 00	REX to access upper XMM reg OVR 128-bit media instruction override
CMPPD \$0,%xmm0,%xmm7	--	--	66	--	0F C2 F8 00	OVR 128-bit media instruction override

CVTSD2SI

CVTSD2SII (%eax),%eax	--	67	F2	--	0f 2d 00	OVR 128-bit media instruction override A32 override: (Addr64) = ZEXT(Addr32)
CVTSD2SII (%r8),%eax	--	--	F2	41	0f 2d 00	OVR 128-bit media instruction override REX to access upper reg.
CVTSD2SII (%rax),%eax	--	--	F2	--	0f 2d 00	OVR 128-bit media instruction override
CVTSD2SII %xmm0,%eax	--	--	F2	--	0f 2d c0	OVR 128-bit media instruction override
CVTSD2SII %xmm15,%eax	--	--	F2	41	0f 2d c7	OVR 128-bit media instruction override REX to access upper XMM reg.
CVTSD2SII %xmm8,%eax	--	--	F2	41	0f 2d c0	OVR 128-bit media instruction override REX to access upper XMM reg.
CVTSD2SII %xmm7,%eax	--	--	F2	--	0f 2d c7	OVR 128-bit media instruction override
CVTSD2SIq (%eax),%rax	--	67	F2	48	0f 2d 00	OVR 128-bit media instruction override A32 override: (Addr64) = ZEXT(Addr32) REX for 64-bit operand size
CVTSD2SIq (%r8),%rax	--	--	F2	49	0f 2d 00	OVR 128-bit media instruction override REX for 64-bit operand size REX to access upper reg.
CVTSD2SIq (%rax),%rax	--	--	F2	48	0f 2d 00	OVR 128-bit media instruction override REX for 64-bit operand size

CVTSD2SIq (%eax),%r8	--	67	F2	4C	0f 2d 00	OVR 128-bit media instruction override A32 override: (Addr64) = ZEXT(Addr32) REX for 64-bit operand size REX to access upper reg.
CVTSD2SIq (%r8),%r8	--	--	F2	4D	0f 2d 00	OVR 128-bit media instruction override REX for 64-bit operand size REX to access upper reg.
CVTSD2SIq (%rax),%r8	--	--	F2	4C	0f 2d 00	OVR 128-bit media instruction override REX for 64-bit operand size REX to access upper reg.
CVTSD2SIq %xmm0,%rax	--	--	F2	48	0f 2d c0	OVR 128-bit media instruction override REX for 64-bit operand size
CVTSD2SIq %xmm15,%r8	--	--	F2	4D	0f 2d c7	OVR 128-bit media instruction override REX for 64-bit operand size REX to access upper XMM reg REX to access upper reg.
CVTSD2SIq %xmm15,%rax	--	--	F2	49	0f 2d c7	OVR 128-bit media instruction override REX for 64-bit operand size REX to access upper XMM reg
CVTSD2SIq %xmm8,%r8	--	--	F2	4D	0f 2d c0	OVR 128-bit media instruction override REX for 64-bit operand size REX to access upper XMM reg REX to access upper reg.
CVTSD2SIq %xmm8,%rax	--	--	F2	49	0f 2d c0	OVR 128-bit media instruction override REX for 64-bit operand size REX to access upper XMM reg
CVTSD2SIq %xmm7,%r8	--	--	F2	4C	0f 2d c7	OVR 128-bit media instruction override REX for 64-bit operand size REX to access upper reg.
CVTSD2SIq %xmm7,%rax	--	--	F2	48	0f 2d c7	OVR 128-bit media instruction override REX for 64-bit operand size
CVTSD2SIq %xmm0,%r8	--	--	F2	4C	0f 2d c0	OVR 128-bit media instruction override REX for 64-bit operand size REX to access upper reg.

CVTTSD2SI

CVTTSD2SII (%eax),%eax	--	67	F2	--	0f 2c 00	OVR 128-bit media instruction override A32 override: (Addr64) = ZEXT(Addr32)
CVTTSD2SII (%r8),%eax	--	--	F2	41	0f 2c 00	OVR 128-bit media instruction override REX to access upper reg.
CVTTSD2SII (%rax),%eax	--	--	F2	--	0f 2c 00	OVR 128-bit media instruction override
CVTTSD2SII %xmm0,%eax	--	--	F2	--	0f 2c c0	OVR 128-bit media instruction override
CVTTSD2SII %xmm15,%eax	--	--	F2	41	0f 2c c7	OVR 128-bit media instruction override REX to access upper XMM reg.
CVTTSD2SII %xmm8,%eax	--	--	F2	41	0f 2c c0	OVR 128-bit media instruction override REX to access upper XMM reg.
CVTTSD2SII %xmm7,%eax	--	--	F2	--	0f 2c c7	OVR 128-bit media instruction override

CVTTSD2Slq (%eax),%rax	--	67	F2	48	0f 2c 00	OVR 128-bit media instruction override A32 override: (Addr64) = ZEXT(Addr32) REX for 64-bit operand size
CVTTSD2Slq (%r8),%rax	--	--	F2	49	0f 2c 00	OVR 128-bit media instruction override REX for 64-bit operand size REX to access upper reg.
CVTTSD2Slq (%rax),%rax	--	--	F2	48	0f 2c 00	OVR 128-bit media instruction override REX for 64-bit operand size
CVTTSD2Slq (%eax),%r8	--	67	F2	4C	0f 2c 00	OVR 128-bit media instruction override A32 override: (Addr64) = ZEXT(Addr32) REX for 64-bit operand size REX to access upper reg.
CVTTSD2Slq (%r8),%r8	--	--	F2	4D	0f 2c 00	OVR 128-bit media instruction override REX for 64-bit operand size REX to access upper reg.
CVTTSD2Slq (%rax),%r8	--	--	F2	4C	0f 2c 00	OVR 128-bit media instruction override REX for 64-bit operand size REX to access upper reg.
CVTTSD2Slq %xmm0,%rax	--	--	F2	48	0f 2c c0	OVR 128-bit media instruction override REX for 64-bit operand size
CVTTSD2Slq %xmm15,%r8	--	--	F2	4D	0f 2c c7	OVR 128-bit media instruction override REX for 64-bit operand size REX to access upper XMM reg REX to access upper reg.
CVTTSD2Slq %xmm15,%rax	--	--	F2	49	0f 2c c7	OVR 128-bit media instruction override REX for 64-bit operand size REX to access upper XMM reg
CVTTSD2Slq %xmm8,%r8	--	--	F2	4D	0f 2c c0	OVR 128-bit media instruction override REX for 64-bit operand size REX to access upper XMM reg REX to access upper reg.
CVTTSD2Slq %xmm8,%rax	--	--	F2	49	0f 2c c0	OVR 128-bit media instruction override REX for 64-bit operand size REX to access upper XMM reg
CVTTSD2Slq %xmm7,%r8	--	--	F2	4C	0f 2c c7	OVR 128-bit media instruction override REX for 64-bit operand size REX to access upper reg.
CVTTSD2Slq %xmm7,%rax	--	--	F2	48	0f 2c c7	OVR 128-bit media instruction override REX for 64-bit operand size
CVTTSD2Slq %xmm0,%r8	--	--	F2	4C	0f 2c c0	OVR 128-bit media instruction override REX for 64-bit operand size REX to access upper reg.

CVTSS2SI

CVTSS2Sll (%eax),%eax	--	67	F3	--	0f 2d 00	OVR 128-bit media instruction override A32 override: (Addr64) = ZEXT(Addr32) Result is sign extended
-----------------------	----	----	----	----	----------	------------------------------------------------------------------------------------------------------------

CVTSS2SII (%r8),%eax	--	--	F3	41	0f 2d 00	Result is sign extended to access upper reg. REX
CVTSS2SII (%rax),%eax	--	--	F3	--	0f 2d 00	OVR 128-bit media instruction override Result is sign extended
CVTSS2SII %xmm0,%eax	--	--	F3	--	0f 2d c0	OVR 128-bit media instruction override Result is sign extended
CVTSS2SII %xmm15,%eax	--	--	F3	41	0f 2d c7	OVR 128-bit media instruction override Result is sign extended REX to access upper XMM reg.
CVTSS2SII %xmm8,%eax	--	--	F3	41	0f 2d c0	OVR 128-bit media instruction override Result is sign extended REX to access upper XMM reg.
CVTSS2SII %xmm7,%eax	--	--	F3	--	0f 2d c7	OVR 128-bit media instruction override Result is sign extended
CVTSS2SIq (%eax),%rax	--	67	F3	48	0f 2d 00	OVR 128-bit media instruction override A32 override: (Addr64) = ZEXT(Addr32) Result is sign extended REX for 64- bit operand size
CVTSS2SIq (%r8),%rax	--	--	F3	49	0f 2d 00	OVR 128-bit media instruction override Result is sign extended REX for 64-bit operand size REX to access upper reg.
CVTSS2SIq (%rax),%rax	--	--	F3	48	0f 2d 00	OVR 128-bit media instruction override Result is sign extended REX for 64-bit operand size
CVTSS2SIq (%eax),%r8	--	67	F3	4C	0f 2d 00	OVR 128-bit media instruction override A32 override: (Addr64) = ZEXT(Addr32) Result is sign extended REX for 64- bit operand size REX to access upper reg.
CVTSS2SIq (%r8),%r8	--	--	F3	4D	0f 2d 00	OVR 128-bit media instruction override Result is sign extended REX for 64-bit operand size REX to access upper reg.
CVTSS2SIq (%rax),%r8	--	--	F3	4C	0f 2d 00	OVR 128-bit media instruction override Result is sign extended REX for 64-bit operand size REX to access upper reg.
CVTSS2SIq %xmm0,%rax	--	--	F3	48	0f 2d c0	OVR 128-bit media instruction override Result is sign extended REX for 64-bit operand size
CVTSS2SIq %xmm15,%r8	--	--	F3	4D	0f 2d c7	OVR 128-bit media instruction override Result is sign extended REX to access upper XMM reg REX to access upper reg.
CVTSS2SIq %xmm15,%rax	--	--	F3	49	0f 2d c7	OVR 128-bit media instruction override Result is sign extended REX for 64-bit operand size REX to access upper XMM reg

CVTSS2Slq %xmm8,%r8	--	--	F3	4D	0f 2d c0	OVR 128-bit media instruction override Result is sign extended REX for 64-bit operand size to access upper XMM reg REX to access upper reg.
CVTSS2Slq %xmm8,%rax	--	--	F3	49	0f 2d c0	OVR 128-bit media instruction override Result is sign extended REX for 64-bit operand size
CVTSS2Slq %xmm7,%r8	--	--	F3	4C	0f 2d c7	OVR 128-bit media instruction override Result is sign extended for 64-bit operand size REX to access upper reg.
CVTSS2Slq %xmm7,%rax	--	--	F3	48	0f 2d c7	OVR 128-bit media instruction override Result is sign extended REX for 64-bit operand size
CVTSS2Slq %xmm0,%r8	--	--	F3	4C	0f 2d c0	OVR 128-bit media instruction override Result is sign extended for 64-bit operand size REX to access upper reg.

CVTTSS2SI

CVTTSS2SII (%eax),%eax	--	67	F3	--	0f 2c 00	OVR 128-bit media instruction override A32 override: (Addr64) = ZEXT(Addr32) Result is sign extended
CVTTSS2SII (%r8),%eax	--	--	F3	41	0f 2c 00	OVR 128-bit media instruction override Result is sign extended to access upper reg.
CVTTSS2SII (%rax),%eax	--	--	F3	--	0f 2c 00	OVR 128-bit media instruction override Result is sign extended
CVTTSS2SII %xmm0,%eax	--	--	F3	--	0f 2c c0	OVR 128-bit media instruction override Result is sign extended
CVTTSS2SII %xmm15,%eax	--	--	F3	41	0f 2c c7	OVR 128-bit media instruction override Result is sign extended to access upper XMM reg.
CVTTSS2SII %xmm8,%eax	--	--	F3	41	0f 2c c0	OVR 128-bit media instruction override Result is sign extended REX to access upper XMM reg.
CVTTSS2SII %xmm7,%eax	--	--	F3	--	0f 2c c7	OVR 128-bit media instruction override Result is sign extended
CVTTSS2Slq (%eax),%rax	--	67	F3	48	0f 2c 00	OVR 128-bit media instruction override A32 override: (Addr64) = ZEXT(Addr32) Result is sign extended REX for 64-bit operand size
CVTTSS2Slq (%r8),%rax	--	--	F3	49	0f 2c 00	OVR 128-bit media instruction override Result is sign extended for 64-bit operand size REX to access upper reg.
CVTTSS2Slq (%rax),%rax	--	--	F3	48	0f 2c 00	OVR 128-bit media instruction override Result is sign extended for 64-bit operand size REX

CVTTSS2Slq (%eax),%r8	--	67	F3	4C	0f 2c 00	OVR 128-bit media instruction override A32 override: (Addr64) = ZEXT(Addr32) Result is sign extended REX for 64- bit operand size REX to access upper reg.
CVTTSS2Slq (%r8),%r8	--	--	F3	4D	0f 2c 00	OVR 128-bit media instruction override Result is sign extended REX for 64-bit operand size REX to access upper reg.
CVTTSS2Slq (%rax),%r8	--	--	F3	4C	0f 2c 00	OVR 128-bit media instruction override Result is sign extended REX for 64-bit operand size REX to access upper reg.
CVTTSS2Slq %xmm0,%rax	--	--	F3	48	0f 2c c0	OVR 128-bit media instruction override Result is sign extended REX for 64-bit operand size
CVTTSS2Slq %xmm15,%r8	--	--	F3	4D	0f 2c c7	OVR 128-bit media instruction override Result is sign extended REX for 64-bit operand size REX to access upper XMM reg REX to access upper reg.
CVTTSS2Slq %xmm15,%rax	--	--	F3	49	0f 2c c7	OVR 128-bit media instruction override Result is sign extended REX for 64-bit operand size REX to access upper XMM reg
CVTTSS2Slq %xmm8,%r8	--	--	F3	4D	0f 2c c0	OVR 128-bit media instruction override Result is sign extended REX for 64-bit operand size REX to access upper XMM reg REX to access upper reg.
CVTTSS2Slq %xmm8,%rax	--	--	F3	49	0f 2c c0	OVR 128-bit media instruction override Result is sign extended REX for 64-bit operand size
CVTTSS2Slq %xmm7,%r8	--	--	F3	4C	0f 2c c7	OVR 128-bit media instruction override Result is sign extended REX for 64-bit operand size REX to access upper reg.
CVTTSS2Slq %xmm7,%rax	--	--	F3	48	0f 2c c7	OVR 128-bit media instruction override Result is sign extended
CVTTSS2Slq %xmm0,%r8	--	--	F3	4C	0f 2c c0	OVR 128-bit media instruction override Result is sign extended REX for 64-bit operand size REX to access upper reg.

CVTSI2SS

CVTSI2SS (%eax),%xmm0	--	67	F3	--	0f 2a 00	OVR 128-bit media instruction override A32 override: (Addr64) = ZEXT(Addr32)
CVTSI2SS (%r8),%xmm0	--	--	F3	41	0f 2a 00	OVR 128-bit media instruction override REX to access upper reg.

CVTSI2SS (%rax),%xmm0	--	--	F3	--	0f 2a 00	OVR 128-bit media instruction override
CVTSI2SS (%eax),%xmm15	--	67	F3	44	0f 2a 38	OVR 128-bit media instruction override A32 override: (Addr64) = ZEXT(Addr32) REX to access upper XMM reg
CVTSI2SS (%r8),%xmm15	--	--	F3	45	0f 2a 38	OVR 128-bit media instruction override REX to access upper XMM reg REX to access upper reg.
CVTSI2SS (%rax),%xmm15	--	--	F3	44	0f 2a 38	OVR 128-bit media instruction override REX to access upper XMM reg
CVTSI2SS (%eax),%xmm8	--	67	F3	44	0f 2a 00	OVR 128-bit media instruction override A32 override: (Addr64) = ZEXT(Addr32)
CVTSI2SS (%r8),%xmm8	--	--	F3	45	0f 2a 00	OVR 128-bit media instruction override REX to access upper XMM reg REX to access upper reg.
CVTSI2SS (%rax),%xmm8	--	--	F3	44	0f 2a 00	OVR 128-bit media instruction override REX to access upper XMM reg
CVTSI2SS (%eax),%xmm7	--	67	F3	--	0f 2a 38	OVR 128-bit media instruction override A32 override: (Addr64) = ZEXT(Addr32)
CVTSI2SS (%r8),%xmm7	--	--	F3	41	0f 2a 38	OVR 128-bit media instruction override REX to access upper reg.
CVTSI2SS (%rax),%xmm7	--	--	F3	--	0f 2a 38	OVR 128-bit media instruction override
CVTSI2SS %eax,%xmm0	--	--	F3	--	0f 2a c0	OVR 128-bit media instruction override
CVTSI2SS %eax,%xmm15	--	--	F3	44	0f 2a f8	OVR 128-bit media instruction override REX to access upper XMM reg
CVTSI2SS %eax,%xmm8	--	--	F3	44	0f 2a c0	OVR 128-bit media instruction override REX to access upper XMM reg
CVTSI2SS %eax,%xmm7	--	--	F3	--	0f 2a f8	OVR 128-bit media instruction override
CVTSI2SS (%eax),%xmm0	--	67	F3	--	0f 2a 00	OVR 128-bit media instruction override A32 override: (Addr64) = ZEXT(Addr32)
CVTSI2SS (%r8),%xmm0	--	--	F3	41	0f 2a 00	OVR 128-bit media instruction override REX to access upper reg.
CVTSI2SS (%rax),%xmm0	--	--	F3	--	0f 2a 00	OVR 128-bit media instruction override
CVTSI2SS (%eax),%xmm15	--	67	F3	44	0f 2a 38	OVR 128-bit media instruction override A32 override: (Addr64) = ZEXT(Addr32) REX to access upper XMM reg
CVTSI2SS (%r8),%xmm15	--	--	F3	45	0f 2a 38	OVR 128-bit media instruction override REX to access upper XMM reg REX to access upper reg.
CVTSI2SS (%rax),%xmm15	--	--	F3	44	0f 2a 38	OVR 128-bit media instruction override REX to access upper XMM reg

CVTSI2SS (%eax),%xmm8	--	67	F3	44	0f 2a 00	OVR 128-bit media instruction override A32 override: (Addr64) = ZEXT(Addr32) REX to access upper XMM reg
CVTSI2SS (%r8),%xmm8	--	--	F3	45	0f 2a 00	OVR 128-bit media instruction override REX to access upper XMM reg REX to access upper reg.
CVTSI2SS (%rax),%xmm8	--	--	F3	44	0f 2a 00	OVR 128-bit media instruction override REX to access upper XMM reg
CVTSI2SS (%eax),%xmm7	--	67	F3	--	0f 2a 38	OVR 128-bit media instruction override A32 override: (Addr64) = ZEXT(Addr32)
CVTSI2SS (%r8),%xmm7	--	--	F3	41	0f 2a 38	OVR 128-bit media instruction override REX to access upper reg.
CVTSI2SS (%rax),%xmm7	--	--	F3	--	0f 2a 38	OVR 128-bit media instruction override
CVTSI2SS %rax,%xmm0	--	--	F3	48	0f 2a c0	OVR 128-bit media instruction override REX for 64-bit operand size
CVTSI2SS %r8,%xmm15	--	--	F3	4D	0f 2a f8	OVR 128-bit media instruction override REX for 64-bit operand size REX to access upper XMM reg REX to access upper reg.
CVTSI2SS %r8,%xmm8	--	--	F3	4D	0f 2a c0	OVR 128-bit media instruction override REX for 64-bit operand size REX to access upper XMM reg REX to access upper reg.
CVTSI2SS %r8,%xmm7	--	--	F3	49	0f 2a f8	OVR 128-bit media instruction override REX for 64-bit operand size REX to access upper reg.
CVTSI2SS %r8,%xmm0	--	--	F3	49	0f 2a c0	OVR 128-bit media instruction override REX for 64-bit operand size REX to access upper reg.
CVTSI2SS %rax,%xmm15	--	--	F3	4C	0f 2a f8	OVR 128-bit media instruction override REX for 64-bit operand size REX to access upper XMM reg
CVTSI2SS %rax,%xmm8	--	--	F3	4C	0f 2a c0	OVR 128-bit media instruction override REX for 64-bit operand size REX to access upper XMM reg
CVTSI2SS %rax,%xmm7	--	--	F3	48	0f 2a f8	OVR 128-bit media instruction override REX for 64-bit operand size

CVTSI2SD

CVTSI2SD (%eax),%xmm0	--	67	F2	--	0F 2A 00	A32 override: (Addr64) = ZEXT(Addr32) OVR 128-bit media instruction override
CVTSI2SD (%r8),%xmm0	--	--	F2	41	0F 2A 00	REX to access upper reg. OVR 128-bit media instruction override
CVTSI2SD (%rax),%xmm0	--	--	F2	--	0F 2A 00	OVR 128-bit media instruction override

CVTSI2SD (%eax),%xmm15	--	67	F2	44	0F 2A 38	REX to access upper XMM reg A32 override: (Addr64) = ZEXT(Addr32) OVR 128-bit media instruction override
CVTSI2SD (%r8),%xmm15	--	--	F2	45	0F 2A 38	REX to access upper XMM reg REX to access upper reg. OVR 128-bit media instruction override
CVTSI2SD (%rax),%xmm15	--	--	F2	44	0F 2A 38	REX to access upper XMM reg OVR 128-bit media instruction override
CVTSI2SD (%eax),%xmm8	--	67	F2	44	0F 2A 00	REX to access upper XMM reg A32 override: (Addr64) = ZEXT(Addr32) OVR 128-bit media instruction override
CVTSI2SD (%r8),%xmm8	--	--	F2	45	0F 2A 00	REX to access upper XMM reg REX to access upper reg. OVR 128-bit media instruction override
CVTSI2SD (%rax),%xmm8	--	--	F2	44	0F 2A 00	REX to access upper XMM reg OVR 128-bit media instruction override
CVTSI2SD (%eax),%xmm7	--	67	F2	--	0F 2A 38	A32 override: (Addr64) = ZEXT(Addr32) OVR 128-bit media instruction override
CVTSI2SD (%r8),%xmm7	--	--	F2	41	0F 2A 38	REX to access upper reg. OVR 128-bit media instruction override
CVTSI2SD (%rax),%xmm7	--	--	F2	--	0F 2A 38	OVR 128-bit media instruction override
CVTSI2SD %eax,%xmm0	--	--	F2	--	0F 2A C0	OVR 128-bit media instruction override
CVTSI2SD %eax,%xmm15	--	--	F2	44	0F 2A F8	REX to access upper XMM reg OVR 128-bit media instruction override
CVTSI2SD %eax,%xmm8	--	--	F2	44	0F 2A C0	REX to access upper XMM reg OVR 128-bit media instruction override
CVTSI2SD %eax,%xmm7	--	--	F2	--	0F 2A F8	OVR 128-bit media instruction override
CVTSI2SD (%eax),%xmm0	--	67	F2	--	0F 2A 00	A32 override: (Addr64) = ZEXT(Addr32) OVR 128-bit media instruction override
CVTSI2SD (%r8),%xmm0	--	--	F2	41	0F 2A 00	REX to access upper reg. OVR 128-bit media instruction override
CVTSI2SD (%rax),%xmm0	--	--	F2	--	0F 2A 00	OVR 128-bit media instruction override
CVTSI2SD (%eax),%xmm15	--	67	F2	44	0F 2A 38	REX to access upper XMM reg A32 override: (Addr64) = ZEXT(Addr32) OVR 128-bit media instruction override
CVTSI2SD (%r8),%xmm15	--	--	F2	45	0F 2A 38	REX to access upper XMM reg REX to access upper reg. OVR 128-bit media instruction override
CVTSI2SD (%rax),%xmm15	--	--	F2	44	0F 2A 38	REX to access upper XMM reg OVR 128-bit media instruction override

CVTSI2SD (%eax),%xmm8	--	67	F2	44	0F 2A 00	REX to access upper XMM reg A32 override: (Addr64) = ZEXT(Addr32) OVR 128-bit media instruction override
CVTSI2SD (%r8),%xmm8	--	--	F2	45	0F 2A 00	REX to access upper XMM reg REX to access upper reg. OVR 128-bit media instruction override
CVTSI2SD (%rax),%xmm8	--	--	F2	44	0F 2A 00	REX to access upper XMM reg OVR 128-bit media instruction override
CVTSI2SD (%eax),%xmm7	--	67	F2	--	0F 2A 38	A32 override: (Addr64) = ZEXT(Addr32) OVR 128-bit media instruction override
CVTSI2SD (%r8),%xmm7	--	--	F2	41	0F 2A 38	REX to access upper reg. OVR 128-bit media instruction override
CVTSI2SD (%rax),%xmm7	--	--	F2	--	0F 2A 38	OVR 128-bit media instruction override
CVTSI2SD %rax,%xmm0	--	--	F2	48	0F 2A C0	REX for 64-bit operand size OVR 128-bit media instruction override
CVTSI2SD %r8,%xmm15	--	--	F2	4D	0F 2A F8	REX to access upper XMM reg REX to access upper reg. REX for 64-bit operand size OVR 128-bit media instruction override
CVTSI2SD %r8,%xmm8	--	--	F2	4D	0F 2A C0	REX to access upper XMM reg REX to access upper reg. REX for 64-bit operand size OVR 128-bit media instruction override
CVTSI2SD %r8,%xmm7	--	--	F2	49	0F 2A F8	REX to access upper reg. REX for 64-bit operand size OVR 128-bit media instruction override
CVTSI2SD %r8,%xmm0	--	--	F2	49	0F 2A C0	REX to access upper reg. REX for 64-bit operand size OVR 128-bit media instruction override
CVTSI2SD %rax,%xmm15	--	--	F2	4C	0F 2A F8	REX to access upper XMM reg REX for 64-bit operand size OVR 128-bit media instruction override
CVTSI2SD %rax,%xmm8	--	--	F2	4C	0F 2A C0	REX to access upper XMM reg REX for 64-bit operand size OVR 128-bit media instruction override
CVTSI2SD %rax,%xmm7	--	--	F2	48	0F 2A F8	REX for 64-bit operand size OVR 128-bit media instruction override

MOVD

MOVD (%eax),%xmm0	--	67	66	--	0F 6E 00	A32 override: (Addr64) = ZEXT(Addr32) Data128 = ZEXT(Data32) OVR 128-bit media instruction override
MOVD (%r8),%xmm0	--	--	66	41	0F 6E 00	REX to access upper reg. Data128 = ZEXT(Data32) OVR 128-bit media instruction override
MOVD (%rax),%xmm0	--	--	66	--	0F 6E 00	Data128 = ZEXT(Data32) OVR 128-bit media instruction override

MOVD (%eax),%xmm15	--	67	66	44	0F 6E 38	REX to access upper XMM reg A32 override: (Addr64) = ZEXT(Addr32) Data128 = ZEXT(Data32) OVR 128-bit media instruction override
MOVD (%r8),%xmm15	--	--	66	45	0F 6E 38	REX to access upper XMM reg REX to access upper reg. Data128 = ZEXT(Data32) OVR 128-bit media instruction override
MOVD (%rax),%xmm15	--	--	66	44	0F 6E 38	REX to access upper XMM reg OVR 128-bit media instruction override
MOVD (%eax),%xmm8	--	67	66	44	0F 6E 00	REX to access upper XMM reg A32 override: (Addr64) = ZEXT(Addr32) Data128 = ZEXT(Data32) OVR 128-bit media instruction override
MOVD (%r8),%xmm8	--	--	66	45	0F 6E 00	REX to access upper XMM reg REX to access upper reg. Data128 = ZEXT(Data32) OVR 128-bit media instruction override
MOVD (%rax),%xmm8	--	--	66	44	0F 6E 00	REX to access upper XMM reg Data128 = ZEXT(Data32) OVR 128-bit media instruction override
MOVD (%eax),%xmm7	--	67	66	--	0F 6E 38	A32 override: (Addr64) = ZEXT(Addr32) Data128 = ZEXT(Data32) OVR 128-bit media instruction override
MOVD (%r8),%xmm7	--	--	66	41	0F 6E 38	REX to access upper reg. Data128 = ZEXT(Data32) OVR 128-bit media instruction override
MOVD (%rax),%xmm7	--	--	66	--	0F 6E 38	Data128 = ZEXT(Data32) OVR 128-bit media instruction override
MOVD %eax,%xmm0	--	--	66	--	0F 6E C0	Data128 = ZEXT(Data32) OVR 128-bit media instruction override
MOVD %eax,%xmm15	--	--	66	44	0F 6E F8	REX to access upper XMM reg Data128 = ZEXT(Data32) OVR 128-bit media instruction override
MOVD %eax,%xmm8	--	--	66	44	0F 6E C0	REX to access upper XMM reg Data128 = ZEXT(Data32) OVR 128-bit media instruction override
MOVD %eax,%xmm7	--	--	66	--	0F 6E F8	Data128 = ZEXT(Data32) OVR 128-bit media instruction override
MOVD %rax,%xmm0	--	--	66	48	0F 6E C0	REX for 64-bit operand size Data128 = ZEXT(Data64) OVR 128-bit media instruction override
MOVD %r8,%xmm15	--	--	66	4D	0F 6E F8	REX to access upper XMM reg REX to access upper reg. OVR 128-bit media instruction override

MOVD %r8,%xmm8	--	--	66	4D	0F 6E C0	REX to access upper XMM reg REX to access upper reg. OVR 128-bit media instruction override
MOVD %r8,%xmm7	--	--	66	49	0F 6E F8	REX to access upper reg. REX for 64-bit operand size Data128 = ZEXT(Data64)
MOVD %r8,%xmm0	--	--	66	49	0F 6E C0	REX to access upper reg. REX for 64-bit operand size OVR 128-bit media instruction override
MOVD %rax,%xmm15	--	--	66	4C	0F 6E F8	REX to access upper XMM reg REX for 64-bit operand size Data128 = ZEXT(Data64) OVR 128-bit media instruction override
MOVD %rax,%xmm8	--	--	66	4C	0F 6E C0	REX to access upper XMM reg REX for 64-bit operand size Data128 = ZEXT(Data64) OVR 128-bit media instruction override
MOVD %rax,%xmm7	--	--	66	48	0F 6E F8	REX for 64-bit operand size Data128 = ZEXT(Data64) OVR 128-bit media instruction override
MOVD %xmm0,(%eax)	--	67	66	--	0F 7E 00	A32 override: (Addr64) = ZEXT(Addr32) OVR 128-bit media instruction override
MOVD %xmm0,(%r8)	--	--	66	41	0F 7E 00	REX to access upper reg. OVR 128-bit media instruction override
MOVD %xmm0,(%rax)	--	--	66	--	0F 7E 00	OVR 128-bit media instruction override
MOVD %xmm15,(%eax)	--	67	66	44	0F 7E 38	REX to access upper XMM reg A32 override: (Addr64) = ZEXT(Addr32) OVR 128-bit media instruction override
MOVD %xmm15,(%r8)	--	--	66	45	0F 7E 38	REX to access upper XMM reg REX to access upper reg. OVR 128-bit media instruction override
MOVD %xmm15,(%rax)	--	--	66	44	0F 7E 38	REX to access upper XMM reg OVR 128-bit media instruction override
MOVD %xmm8,(%eax)	--	67	66	44	0F 7E 00	REX to access upper XMM reg A32 override: (Addr64) = ZEXT(Addr32) OVR 128-bit media instruction override
MOVD %xmm8,(%r8)	--	--	66	45	0F 7E 00	REX to access upper XMM reg REX to access upper reg. OVR 128-bit media instruction override
MOVD %xmm8,(%rax)	--	--	66	44	0F 7E 00	REX to access upper XMM reg OVR 128-bit media instruction override
MOVD %xmm7,(%eax)	--	67	66	--	0F 7E 38	A32 override: (Addr64) = ZEXT(Addr32) OVR 128-bit media instruction override
MOVD %xmm7,(%r8)	--	--	66	41	0F 7E 38	REX to access upper reg. OVR 128-bit media instruction override

MOVD %xmm7,(%rax)	--	--	66	--	0F 7E 38	OVR 128-bit media instruction override
MOVD %xmm0,%eax	--	--	66	--	0F 7E C0	OVR 128-bit media instruction override
MOVD %xmm15,%eax	--	--	66	44	0F 7E F8	REX to access upper XMM reg OVR 128-bit media instruction override
MOVD %xmm8,%eax	--	--	66	44	0F 7E C0	REX to access upper XMM reg OVR 128-bit media instruction override
MOVD %xmm7,%eax	--	--	66	--	0F 7E F8	OVR 128-bit media instruction override
MOVD %xmm0,%rax	--	--	66	48	0F 7E C0	REX for 64-bit operand size OVR 128-bit media instruction override
MOVD %xmm15,%r8	--	--	66	4D	0F 7E F8	REX to access upper XMM reg REX to access upper reg. OVR 128-bit media instruction override
MOVD %xmm15,%rax	--	--	66	4C	0F 7E F8	REX to access upper XMM reg REX for 64-bit operand size OVR 128-bit media instruction override
MOVD %xmm8,%r8	--	--	66	4D	0F 7E C0	REX to access upper XMM reg REX to access upper reg. OVR 128-bit media instruction override
MOVD %xmm8,%rax	--	--	66	4C	0F 7E C0	REX to access upper XMM reg REX for 64-bit operand size OVR 128-bit media instruction override
MOVD %xmm7,%r8	--	--	66	49	0F 7E F8	REX to access upper reg. REX for 64-bit operand size OVR 128-bit media instruction override
MOVD %xmm7,%rax	--	--	66	48	0F 7E F8	REX for 64-bit operand size OVR 128-bit media instruction override
MOVD %xmm0,%r8	--	--	66	49	0F 7E C0	REX to access upper reg. REX for 64-bit operand size OVR 128-bit media instruction override
MOVD %rax,%xmm0	--	--	66	48	0F 6E C0	REX for 64-bit operand size Data128 = ZEXT(Data64) OVR 128-bit media instruction override
MOVD %r8,%xmm15	--	--	66	4D	0F 6E F8	REX to access upper XMM reg REX to access upper reg. Data128 = ZEXT(Data64) OVR 128-bit media instruction override
MOVD %r8,%xmm8	--	--	66	4D	0F 6E C0	REX to access upper XMM reg REX to access upper reg. Data128 = ZEXT(Data64) OVR 128-bit media instruction override
MOVD %r8,%xmm7	--	--	66	49	0F 6E F8	REX to access upper reg. REX for 64-bit operand size Data128 = ZEXT(Data64) OVR 128-bit media instruction override
MOVD %r8,%xmm0	--	--	66	49	0F 6E C0	REX to access upper reg. REX for 64-bit operand size Data128 = ZEXT(Data64) OVR 128-bit media instruction override

MOVD %rax,%xmm15	--	--	66	4C	0F 6E F8	REX to access upper XMM reg REX for 64-bit operand size Data128 = ZEXT(Data64) OVR 128-bit media instruction override
MOVD %rax,%xmm8	--	--	66	4C	0F 6E C0	REX to access upper XMM reg REX for 64-bit operand size Data128 = ZEXT(Data64) OVR 128-bit media instruction override
MOVD %rax,%xmm7	--	--	66	48	0F 6E F8	REX for 64-bit operand size Data128 = ZEXT(Data64) OVR 128-bit media instruction override
MOVD %xmm0,%rax	--	--	66	48	0F 7E C0	REX for 64-bit operand size OVR 128-bit media instruction override
MOVD %xmm15,%r8	--	--	66	4D	0F 7E F8	REX to access upper XMM reg REX to access upper reg. OVR 128-bit media instruction override
MOVD %xmm15,%rax	--	--	66	4C	0F 7E F8	REX to access upper XMM reg REX for 64-bit operand size OVR 128-bit media instruction override
MOVD %xmm8,%r8	--	--	66	4D	0F 7E C0	REX to access upper XMM reg REX to access upper reg. OVR 128-bit media instruction override
MOVD %xmm8,%rax	--	--	66	4C	0F 7E C0	REX to access upper XMM reg REX for 64-bit operand size OVR 128-bit media instruction override
MOVD %xmm7,%r8	--	--	66	49	0F 7E F8	REX to access upper reg. REX for 64-bit operand size OVR 128-bit media instruction override
MOVD %xmm7,%rax	--	--	66	48	0F 7E F8	REX for 64-bit operand size OVR 128-bit media instruction override
MOVD %xmm0,%r8	--	--	66	49	0F 7E C0	REX to access upper reg. REX for 64-bit operand size OVR 128-bit media instruction override

MOVQ

MOVQ (%eax),%xmm0	--	67	F3	--	0F 7E 00	A32 override: (Addr64) = ZEXT(Addr32) Data128 = ZEXT(Data64) OVR 128-bit media instruction override
MOVQ (%r8),%xmm0	--	--	F3	41	0F 7E 00	REX to access upper reg. Data128 = ZEXT(Data64) OVR 128-bit media instruction override
MOVQ (%rax),%xmm0	--	--	F3	--	0F 7E 00	Data128 = ZEXT(Data64) OVR 128-bit media instruction override
MOVQ (%eax),%xmm15	--	67	F3	44	0F 7E 38	REX to access upper XMM reg A32 override: (Addr64) = ZEXT(Addr32) Data128 = ZEXT(Data64) OVR 128-bit media instruction override

MOVQ (%r8),%xmm15	--	--	F3	45	0F 7E 38	REX to access upper XMM reg REX to access upper reg. Data128 = ZEXT(Data64) OVR 128-bit media instruction override
MOVQ (%rax),%xmm15	--	--	F3	44	0F 7E 38	REX to access upper XMM reg Data128 = ZEXT(Data64) OVR 128-bit media instruction override
MOVQ (%eax),%xmm8	--	67	F3	44	0F 7E 00	REX to access upper XMM reg A32 override: (Addr64) = ZEXT(Addr32) Data128 = ZEXT(Data64) OVR 128-bit media instruction override
MOVQ (%r8),%xmm8	--	--	F3	45	0F 7E 00	REX to access upper XMM reg REX to access upper reg. Data128 = ZEXT(Data64) OVR 128-bit media instruction override
MOVQ (%rax),%xmm8	--	--	F3	44	0F 7E 00	REX to access upper XMM reg Data128 = ZEXT(Data64) OVR 128-bit media instruction override
MOVQ (%eax),%xmm7	--	67	F3	--	0F 7E 38	A32 override: (Addr64) = ZEXT(Addr32) Data128 = ZEXT(Data64) OVR 128-bit media instruction override
MOVQ (%r8),%xmm7	--	--	F3	41	0F 7E 38	REX to access upper reg. Data128 = ZEXT(Data64) OVR 128-bit media instruction override
MOVQ (%rax),%xmm7	--	--	F3	--	0F 7E 38	Data128 = ZEXT(Data64) OVR 128-bit media instruction override
MOVQ %xmm0,%xmm0	--	--	F3	--	0F 7E C0	OVR 128-bit media instruction override
MOVQ %xmm15,%xmm15	--	--	F3	45	0F 7E FF	REX to access upper XMM reg OVR 128-bit media instruction override
MOVQ %xmm15,%xmm8	--	--	F3	45	0F 7E C7	REX to access upper XMM reg OVR 128-bit media instruction override
MOVQ %xmm15,%xmm7	--	--	F3	41	0F 7E FF	REX to access upper XMM reg OVR 128-bit media instruction override
MOVQ %xmm15,%xmm0	--	--	F3	41	0F 7E C7	REX to access upper XMM reg OVR 128-bit media instruction override
MOVQ %xmm8,%xmm15	--	--	F3	45	0F 7E F8	REX to access upper XMM reg OVR 128-bit media instruction override
MOVQ %xmm8,%xmm8	--	--	F3	45	0F 7E C0	REX to access upper XMM reg OVR 128-bit media instruction override
MOVQ %xmm8,%xmm7	--	--	F3	41	0F 7E F8	REX to access upper XMM reg OVR 128-bit media instruction override
MOVQ %xmm8,%xmm0	--	--	F3	41	0F 7E C0	REX to access upper XMM reg OVR 128-bit media instruction override
MOVQ %xmm7,%xmm15	--	--	F3	44	0F 7E FF	REX to access upper XMM reg OVR 128-bit media instruction override
MOVQ %xmm7,%xmm8	--	--	F3	44	0F 7E C7	REX to access upper XMM reg OVR 128-bit media instruction override

MOVQ %xmm7,%xmm7	--	--	F3	--	0F 7E FF	OVR 128-bit media instruction override
MOVQ %xmm7,%xmm0	--	--	F3	--	0F 7E C7	OVR 128-bit media instruction override
MOVQ %xmm0,%xmm15	--	--	F3	44	0F 7E F8	REX to access upper XMM reg OVR 128-bit media instruction override
MOVQ %xmm0,%xmm8	--	--	F3	44	0F 7E C0	REX to access upper XMM reg OVR 128-bit media instruction override
MOVQ %xmm0,%xmm7	--	--	F3	--	0F 7E F8	OVR 128-bit media instruction override
MOVQ %xmm0,(%eax)	--	67	66	--	0F D6 00	A32 override: (Addr64) = ZEXT(Addr32) OVR 128-bit media instruction override
MOVQ %xmm0,(%r8)	--	--	66	41	0F D6 00	REX to access upper reg. OVR 128-bit media instruction override
MOVQ %xmm0,(%rax)	--	--	66	--	0F D6 00	OVR 128-bit media instruction override
MOVQ %xmm15,(%eax)	--	67	66	44	0F D6 38	REX to access upper XMM reg A32 override: (Addr64) = ZEXT(Addr32) OVR 128-bit media instruction override
MOVQ %xmm15,(%r8)	--	--	66	45	0F D6 38	REX to access upper reg. OVR 128-bit media instruction override
MOVQ %xmm15,(%rax)	--	--	66	44	0F D6 38	REX to access upper XMM reg OVR 128-bit media instruction override
MOVQ %xmm8,(%eax)	--	67	66	44	0F D6 00	REX to access upper XMM reg A32 override: (Addr64) = ZEXT(Addr32) OVR 128-bit media instruction override
MOVQ %xmm8,(%r8)	--	--	66	45	0F D6 00	REX to access upper XMM reg REX to access upper reg. OVR 128-bit media instruction override
MOVQ %xmm8,(%rax)	--	--	66	44	0F D6 00	REX to access upper XMM reg OVR 128-bit media instruction override
MOVQ %xmm7,(%eax)	--	67	66	--	0F D6 38	A32 override: (Addr64) = ZEXT(Addr32) OVR 128-bit media instruction override
MOVQ %xmm7,(%r8)	--	--	66	41	0F D6 38	REX to access upper reg. OVR 128-bit media instruction override

64-bit MMX CVTPD2PI

CVTPD2PI (%eax),%mm0	--	67	66	--	0F 2D 00	A32 override: (Addr64) = ZEXT(Addr32) OVR 128-bit media instruction override
CVTPD2PI (%r8),%mm0	--	--	66	41	0F 2D 00	REX to access upper reg. OVR 128-bit media instruction override
CVTPD2PI (%rax),%mm0	--	--	66	--	0F 2D 00	OVR 128-bit media instruction override
CVTPD2PI (%eax),%mm7	--	67	66	--	0F 2D 38	A32 override: (Addr64) = ZEXT(Addr32) OVR 128-bit media instruction override
CVTPD2PI (%r8),%mm7	--	--	66	41	0F 2D 38	REX to access upper reg. OVR 128-bit media instruction override
CVTPD2PI (%rax),%mm7	--	--	66	--	0F 2D 38	OVR 128-bit media instruction override

CVTPD2PI %xmm0,%mm0	--	--	66	--	0F 2D C0	OVR 128-bit media instruction override
CVTPD2PI %xmm15,%mm7	--	--	66	41	0F 2D FF	REX to access upper XMM reg OVR 128-bit media instruction override
CVTPD2PI %xmm15,%mm0	--	--	66	41	0F 2D C7	REX to access upper XMM reg OVR 128-bit media instruction override
CVTPD2PI %xmm8,%mm7	--	--	66	41	0F 2D F8	REX to access upper XMM reg OVR 128-bit media instruction override
CVTPD2PI %xmm8,%mm0	--	--	66	41	0F 2D C0	REX to access upper XMM reg OVR 128-bit media instruction override
CVTPD2PI %xmm7,%mm7	--	--	66	--	0F 2D FF	OVR 128-bit media instruction override
CVTPD2PI %xmm7,%mm0	--	--	66	--	0F 2D C7	OVR 128-bit media instruction override
CVTPD2PI %xmm0,%mm7	--	--	66	--	0F 2D F8	OVR 128-bit media instruction override

MOVD

MOVD (%eax),%mm0	--	67	--	--	0F 6E 00	A32 override: (Addr64) = ZEXT(Addr32) Data64 = ZEXT(Data32)
MOVD (%r8),%mm0	--	--	--	41	0F 6E 00	REX to access upper reg. Data64 = ZEXT(Data32)
MOVD (%rax),%mm0	--	--	--	--	0F 6E 00	Data64 = ZEXT(Data32)
MOVD (%eax),%mm7	--	67	--	--	0F 6E 38	A32 override: (Addr64) = ZEXT(Addr32) Data64 = ZEXT(Data32)
MOVD (%r8),%mm7	--	--	--	41	0F 6E 38	REX to access upper reg. Data64 = ZEXT(Data32)
MOVD (%rax),%mm7	--	--	--	--	0F 6E 38	Data64 = ZEXT(Data32)
MOVD %eax,%mm0	--	--	--	--	0F 6E C0	Data64 = ZEXT(Data32)
MOVD %eax,%mm7	--	--	--	--	0F 6E F8	Data64 = ZEXT(Data32)
MOVD %mm0,(%eax)	--	67	--	--	0F 7E 00	A32 override: (Addr64) = ZEXT(Addr32)
MOVD %mm0,(%r8)	--	--	--	41	0F 7E 00	REX to access upper reg.
MOVD %mm0,(%rax)	--	--	--	--	0F 7E 00	
MOVD %mm7,(%eax)	--	67	--	--	0F 7E 38	A32 override: (Addr64) = ZEXT(Addr32)
MOVD %mm7,(%r8)	--	--	--	41	0F 7E 38	REX to access upper reg.
MOVD %mm7,(%rax)	--	--	--	--	0F 7E 38	
MOVD %mm0,%eax	--	--	--	--	0F 7E C0	
MOVD %mm7,%eax	--	--	--	--	0F 7E F8	
MOVD %rax,%mm0	--	--	--	48	0F 6E C0	REX for 64-bit operand size
MOVD %r8,%mm7	--	--	--	49	0F 6E F8	REX for 64-bit operand size REX to access upper reg.
MOVD %r8,%mm0	--	--	--	49	0F 6E C0	REX for 64-bit operand size REX to access upper reg.
MOVD %rax,%mm7	--	--	--	48	0F 6E F8	REX for 64-bit operand size
MOVD %mm0,%rax	--	--	--	48	0F 7E C0	REX for 64-bit operand size
MOVD %mm7,%r8	--	--	--	49	0F 7E F8	REX for 64-bit operand size REX to access upper reg.

MOVD %mm7,%rax	--	--	--	48	0F 7E F8	REX for 64-bit operand size
MOVD %mm0,%r8	--	--	--	49	0F 7E C0	REX for 64-bit operand size REX to access upper reg.

MOVQ

MOVQ (%eax),%mm0	--	67	--	--	0F 6F 00	A32 override: (Addr64) = ZEXT(Addr32)
MOVQ (%r8),%mm0	--	--	--	41	0F 6F 00	REX to access upper reg.
MOVQ (%rax),%mm0	--	--	--	--	0F 6F 00	
MOVQ (%eax),%mm7	--	67	--	--	0F 6F 38	A32 override: (Addr64) = ZEXT(Addr32)
MOVQ (%r8),%mm7	--	--	--	41	0F 6F 38	REX to access upper reg.
MOVQ (%rax),%mm7	--	--	--	--	0F 6F 38	
MOVQ %mm0,%mm0	--	--	--	--	0F 7F C0	
MOVQ %mm7,%mm7	--	--	--	--	0F 7F FF	
MOVQ %mm7,%mm0	--	--	--	--	0F 7F F8	
MOVQ %mm0,%mm7	--	--	--	--	0F 7F C7	
MOVQ %mm0,(%eax)	--	67	--	--	0F 7F 00	A32 override: (Addr64) = ZEXT(Addr32)
MOVQ %mm0,(%r8)	--	--	--	41	0F 7F 00	REX to access upper reg.
MOVQ %mm0,(%rax)	--	--	--	--	0F 7F 00	
MOVQ %mm7,(%eax)	--	67	--	--	0F 7F 38	A32 override: (Addr64) = ZEXT(Addr32)
MOVQ %mm7,(%r8)	--	--	--	41	0F 7F 38	REX to access upper reg.
MOVQ %mm7,(%rax)	--	--	--	--	0F 7F 38	

X87

FADDP

FADDP %st(0),%st(0)	--	--	--	--	DE C0	
FADDP %st(0),%st(7)	--	--	--	--	DE C7	
FADDP	--	--	--	--	DE C1	

FDIV

FDIV (%eax)	--	67	--	--	D8 30	A32 override: (Addr64) = ZEXT(Addr32)
FDIV (%r8)	--	--	--	41	D8 30	REX to access upper reg.
FDIV (%rax)	--	--	--	--	D8 30	
FDIV (%eax)	--	67	--	--	DC 30	A32 override: (Addr64) = ZEXT(Addr32)
FDIV (%r8)	--	--	--	41	DC 30	REX to access upper reg.
FDIV (%rax)	--	--	--	--	DC 30	
FDIV %st(1),%st(0)	--	--	--	--	DC F9	
FDIV %st(0),%st(0)	--	--	--	--	DC F8	
FDIV %st(0),%st(7)	--	--	--	--	DC FF	
FDIV %st(0),%st(0)	--	--	--	--	DC F8	
FDIV %st(7),%st(0)	--	--	--	--	D8 F7	



**Stack
Operations**
POP

POPw %ax	66	--	--	--	58	O16 for 16-bit operand size
POPl %eax	--	--	--	--	ILLEGAL	can't have 32-bit stack operands
POPq %rax	--	--	--	--	58	
POPq %r8	--	--	--	41	58	REX to access upper reg. REX for 64-bit operand size
POPq (%eax)	--	67	--	--	8F 00	A32 override: (Addr64) = ZEXT(Addr32)
POPq (%r8)	--	--	--	41	8F 00	REX to access upper reg.
POPq (%rax)	--	--	--	--	8F 00	
POP %fs	--	--	--	--	0F A1	
POPF	66	--	--	--	9D	O16 for 16-bit operand size
POPFD	--	--	--	--	ILLEGAL	can't have 32-bit stack operands
POPfq	--	--	--	--	9D	

PUSH

PUSH %fs	--	--	--	--	0F A0	
PUSHb \$0	--	--	--	--	6A 00	
PUSHw \$0x7000	66	--	--	--	68 00 70	O16 for 16-bit operand size
Pushw %ax	66	--	--	--	50	O16 for 16-bit operand size
Pushl %eax	--	--	--	--	ILLEGAL	can't have 32-bit stack operands
PUSHq %rax	--	--	--	--	50	
PUSHq %r8	--	--	--	41	50	REX to access upper reg. REX for 64-bit operand size
PUSHq (%eax)	--	67	--	--	FF 30	A32 override: (Addr64) = ZEXT(Addr32)
PUSHq (%r8)	--	--	--	41	FF 30	REX to access upper reg.
PUSHq (%rax)	--	--	--	--	FF 30	
PUSHF	66	--	--	--	9C	O16 for 16-bit operand size
PUSHFD	--	--	--	--	ILLEGAL	can't have 32-bit stack operands
PUSHfq	--	--	--	--	9C	

MMX/XMM/x

87 State

FNSAVE	FNSAVE .(%RIP)	--	--	--	--	DD 35 FA FF FF FF	
FRSTOR	FRSTOR .(%RIP)	--	--	--	--	DD 25 FA FF FF FF	
FSAVE	FSAVE .(%RIP)	--	--	--	--	9B DD 35 FA FF FF FF	
FXRSTOR	FXRSTOR .(%RIP)	--	--	--	--	0F AE 0D F9 FF FF FF	

FXSAVE	FXSAVE .(%RIP)	--	--	--	--	0F AE 05 F9 FF FF FF	
EMMS	EMMS	--	--	--	--	0F 77	
FEMMS	FEMMS	--	--	--	--	0F 0E	

LEA calculation

LEA	LEAw %fs:(%eax),%ax	66	67	64	--	8D	O16 for 16-bit operand size A32 override: (Addr64) = ZEXT(Addr32) OVR FS segment override
	LEAw %fs:(%rax),%ax	66	--	64	--	8D	O16 for 16-bit operand size OVR FS segment override
	LEAl %fs:(%eax),%eax	--	67	64	--	8D	A32 override: (Addr64) = ZEXT(Addr32)
	LEAl %fs:(%rax),%eax	--	--	64	--	8D	OVR FS segment override
	LEAq %fs:(%eax),%rax	--	67	64	48	8D	REX for 64-bit operand size A32 override: (Addr64) = ZEXT(Addr32) OVR FS segment override
	LEAq %fs:(%rax),%rax	--	--	64	48	8D	REX for 64-bit operand size OVR FS segment override

MISC System Instructions

CLFLUSH	CLFLUSH (%eax)	--	67	--	--	0F AE 38	A32 override: (Addr64) = ZEXT(Addr32)
	CLFLUSH (%r8)	--	--	--	41	0F AE 38	REX to access upper reg.
	CLFLUSH (%rax)	--	--	--	--	0F AE 38	

INVD	INVD	--	--	--	--	0F 08	
------	------	----	----	----	----	-------	--

INVLPG	INVLPG (%eax)	--	67	--	--	0F 01 38	A32 override: (Addr64) = ZEXT(Addr32)
	INVLPG (%r8)	--	--	--	41	0F 01 38	REX to access upper reg.
	INVLPG (%rax)	--	--	--	--	0F 01 38	
	INVLPG (%eax)	--	67	--	--	0F 01 38	A32 override: (Addr64) = ZEXT(Addr32)
	INVLPG (%r8)	--	--	--	41	0F 01 38	REX to access upper reg.
	INVLPG (%rax)	--	--	--	--	0F 01 38	
	INVLPG (%eax)	--	67	--	--	0F 01 38	A32 override: (Addr64) = ZEXT(Addr32)
	INVLPG (%r8)	--	--	--	41	0F 01 38	REX to access upper reg.

INVLPG (%rax)	--	--	--	--	0F 01 38	
---------------	----	----	----	----	----------	--

LAR	LARw (%eax),%ax	66	67	--	--	0F 02 00	O16 for 16-bit operand size A32 override: (Addr64) = ZEXT(Addr32)
	LARw (%r8),%ax	66	--	--	41	0F 02 00	REX to access upper reg. O16 for 16-bit operand size
	LARw (%rax),%ax	66	--	--	--	0F 02 00	O16 for 16-bit operand size
	LARw %ax,%ax	66	--	--	--	0F 02 C0	O16 for 16-bit operand size
	LARI (%eax),%eax	--	67	--	--	0F 02 00	A32 override: (Addr64) = ZEXT(Addr32)
	LARI (%r8),%eax	--	--	--	41	0F 02 00	REX to access upper reg.
	LARI (%rax),%eax	--	--	--	--	0F 02 00	
	LARI %eax,%eax	--	--	--	--	0F 02 C0	

LGDT	LGDT %fs:0	--	--	64	--	0F 01 14 25 00 00 00 00	OVR FS segment override
	LIDT %fs:0	--	--	64	--	0F 01 1C 25 00 00 00 00	OVR FS segment override
LLDT	LLDTw (%eax)	--	67	--	--	0F 00 10	A32 override: (Addr64) = ZEXT(Addr32)
	LLDTw (%r8)	--	--	--	41	0F 00 10	REX to access upper reg.
	LLDTw (%rax)	--	--	--	--	0F 00 10	
	LLDTw %ax	--	--	--	--	0F 00 D0	

SGDT	SGDT (%eax)	--	67	--	--	0F 01 00	A32 override: (Addr64) = ZEXT(Addr32)
	SGDT (%r8)	--	--	--	41	0F 01 00	REX to access upper reg.
	SGDT (%rax)	--	--	--	--	0F 01 00	
	SGDT (%eax)	--	67	--	--	0F 01 00	A32 override: (Addr64) = ZEXT(Addr32)
	SGDT (%r8)	--	--	--	41	0F 01 00	REX to access upper reg.
	SGDT (%rax)	--	--	--	--	0F 01 00	
SIDT	SIDT (%eax)	--	67	--	--	0F 01 08	A32 override: (Addr64) = ZEXT(Addr32)
	SIDT (%r8)	--	--	--	41	0F 01 08	REX to access upper reg.
	SIDT (%rax)	--	--	--	--	0F 01 08	
	SIDT (%eax)	--	67	--	--	0F 01 08	A32 override: (Addr64) = ZEXT(Addr32)
	SIDT (%r8)	--	--	--	41	0F 01 08	REX to access upper reg.
	SIDT (%rax)	--	--	--	--	0F 01 08	
SLDT	SLDTw (%eax)	66	67	--	--	0F 00 00	A32 override: (Addr64) = ZEXT(Addr32) O16 for 16-bit operand size
	SLDTw (%r8)	66	--	--	41	0F 00 00	REX to access upper reg. O16 for 16-bit operand size

SLDTw (%rax)	66	--	--	--	0F 00 00	O16 for 16-bit operand size
SLDTw %ax	66	--	--	--	0F 00 C0	O16 for 16-bit operand size
SLDT (%eax)	--	67	--	--	0F 00 00	A32 override: (Addr64) = ZEXT(Addr32)
SLDT (%r8)	--	--	--	41	0F 00 00	REX to access upper reg.
SLDT (%rax)	--	--	--	--	0F 00 00	
SLDT %eax	--	--	--	--	0F 00 C0	

SWAPGS
IO
OUT

SWAPGS	--	--	--	--	0F 01 F8	
--------	----	----	----	----	----------	--

OUT %al,\$0	--	--	--	--	E6 00	
OUT %ax,\$0	66	--	--	--	E7 00	O16 for 16-bit operand size
OUT %eax,\$0	--	--	--	--	E7 00	
OUTb %al,%dx	--	--	--	--	EE	
OUTw %ax,%dx	66	--	--	--	EF	O16 for 16-bit operand size
OUTl %eax,%dx	--	--	--	--	EF	
OUTSB	--	--	--	--	6E	Output data p't to by (RSI)
OUTSB ??	--	--	67	--	6E	OVR Out data p'td to by ZEXT(eSI) to port p'td to by DX
OUTSI	--	--	--	--	6F	Output data p't to by (RSI)
OUTSI ??	--	--	67	--	6F	OVR Out data p'td to by ZEXT(eSI) to port p'td to by DX
OUTSW	66	--	--	--	6F	O16 for 16-bit operand size
OUTSW ??	66	--	67	--	6F	O16 for 16-bit operand size OVR Out data p'td to by ZEXT(eSI) to port p'td to by DX
	--	--	--	--		
INb \$0,%al	--	--	--	--	E4 00	
INw \$0,%ax	66	--	--	--	E5 00	O16 for 16-bit operand size
INI \$0,%eax	--	--	--	--	E5 00	
INb %dx,%al	--	--	--	--	EC	
INw %dx,%ax	66	--	--	--	ED	O16 for 16-bit operand size
INI %dx,%eax	--	--	--	--	ED	
INSB	--	--	--	--	6C	Input data p'td to by (RSI)
INSB ??	--	--	67	--	6C	OVR Input data p'td to by ZEXT(eSI) to port p'td to by DX
INSI	--	--	--	--	6D	Input data p'td to by (RSI)
INSI ??	--	--	67	--	6D	OVR Input data p'td to by ZEXT(eSI) to port p'td to by DX
INSW	66	--	--	--	6D	O16 for 16-bit operand size
INSW ??	66	--	67	--	6D	O16 for 16-bit operand size OVR Input data p'td to by ZEXT(eSI) to port p'td to by DX

IN